Optimal Rates for Random Order Online Optimization

Anonymous Author(s)
Affiliation
Address
email

Abstract

We study online convex optimization in the random order model, recently proposed by Garber et al. [8], where the loss functions may be chosen by an adversary, but are then presented to the online algorithm in a uniformly random order. Focusing on the scenario where the cumulative loss function is (strongly) convex, yet individual loss functions are smooth but might be non-convex, we give algorithms that achieve the optimal bounds and significantly outperform the results of Garber et al. [8], completely removing the dimension dependence and improving their scaling with respect to the strong convexity parameter. Our analysis relies on novel connections between algorithmic stability and generalization for sampling without-replacement analogous to those studied in the with-replacement i.i.d. setting, as well as on a refined average stability analysis of stochastic gradient descent.

1 Introduction

Online convex optimization [24, 12] studies the iterative process of decision making as data arrives in an online fashion. The model posits a game of T rounds, where in each round the learner chooses a decision w_t from a convex set $W \subseteq \mathbb{R}^d$, after which she observes a loss function $f_t : W \to \mathbb{R}$, and incurs loss $f_t(w_t)$. The learner's objective is to minimize her regret, defined as her cumulative loss minus that of the best decision in hindsight $w^* = \arg\min_{w \in W} \sum_{t=1}^{T} f_t(w)$. In the prototypical setting, the individual loss functions are assumed to be convex and adversarially chosen by an opponent—commonly known as nature or the adversary—who has knowledge of the learner's algorithm. While this setup is fundamental enough to accommodate a diverse set of applications (see, e.g., [12]), studying variants of the basic model promotes modeling flexibility, and further broadens the set of problems to which optimization techniques may be applied.

Recently, Garber et al. [8] consider relaxing the convexity assumption by requiring that only *on average* the loss is (strongly) convex—a property the authors refer to as cumulative (strong) convexity—but do not require that the losses are convex individually. It is well known (e.g., [4]) that under these assumptions, if the losses are sampled i.i.d. from some distribution, stochastic gradient descent (SGD) obtains the optimal $O(\log T)$ regret in expectation. However, as it turns out, in the fully adversarial model the cumulative strong convexity assumption is too weak: Garber et al. [8] show that in this case there is a linear regret lower bound. Consequently, they propose the *random order* model, where T losses are chosen adversarially but then revealed to the learner in uniformly random order. Within this model, under the relaxed convexity assumption Garber et al. [8] obtain sub-linear regret for a number of specialized settings, differing in their assumptions on the structure of the individual loss functions. In the most general case (the one we consider in this paper), they prove online gradient descent obtains regret $O((dG^2/\lambda^3)\log T)$ w.h.p. for G-Lipschitz λ-cumulative-strongly convex losses.

It is informative to compare the random order model with the i.i.d. stochastic case, where on every round a new loss is sampled uniformly and *independently* from the set of losses, that is, *with-replacement*. By contrast, the random order model specifies that on every round a new loss is sampled uniformly *without-replacement*, an in particular *not independently*. Concretely, let $\mathcal{L}$ be

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

an arbitrary set of T smooth and Lipschitz continuous loss functions. Set $F(w) := \frac{1}{T}\sum_{f\in\mathcal{L}} f(w)$, and assume F is λ-strongly-convex over W. The learner's goal is to minimize her regret on the loss sequence $f_1, \ldots, f_T$ obtained from a uniformly random ordering of $\mathcal{L}$. As noted previously, if the losses f_t were drawn i.i.d., SGD obtains the optimal $O(\log T)$ regret even though the losses are not individually convex. In a nutshell, when losses are i.i.d., the gradients used in the SGD update are conditionally unbiased estimates of the gradient of the average (strongly convex) loss, hence the optimal regret is achieved in expectation. The difficulty in the random order model stems from the fact that random order gradients are not conditionally unbiased; given any set of past losses $f_1, \ldots, f_{t-1}$, the next loss is uniform over the complement $\mathcal{L} \setminus \{f_1, \ldots, f_{t-1}\}$, and thus $\nabla f_t(w_t)$ is biased.

To overcome this complication, Garber et al. [8] work via the uniform convergence route, and build on concentration bounds applied to Hessians of the losses. As a result, they achieve suboptimal bounds, particularly in the general case where a dimension factor is introduced by a discretization argument necessary to ensure convergence over the entire domain. Here we choose a different strategy, and draw connections to notions of algorithmic stability and generalization studied in statistical learning theory. Our approach introduces significant improvements compared to prior work, and achieves regret bounds optimal up to additive factors.

1.1 Our results

We present and analyze two algorithms for random order online optimization, the first of which obtains the optimal regret up to additive factors. Let $f_1, \ldots, f_T$ be a random order sequence of G-Lipschitz, β-smooth losses, where $f_t : W \to \mathbb{R}$ for all $t \le T$. Assume the domain $W \subset \mathbb{R}^d$ is convex, and has diameter bounded by D. Further, assume the average loss $\frac{1}{T}\sum_{t=1}^{T} f_t$ is λ-strongly convex. We prove;

Theorem (informal)**.** *There exists an algorithm (Algorithm 1) for random order online optimization that obtains regret of* $O\left((G^2/\lambda)\log T\right)$ *in expectation.*

Although the above result matches the optimal result for the *individually* strongly convex setting (up to additive factors), Algorithm 1 requires memory linear in T. This disadvantage motivates another algorithm, which trades off an extra factor of $\kappa = \beta/\lambda$ in the regret for lower memory requirements.

Theorem (informal)**.** *There exists an algorithm (Algorithm 2) for random order online optimization that requires memory linear in d, and obtains regret of* $O\left((G^2/\lambda^2)\log T\right)$ *in expectation.*

Big-O notation in both theorems hides additive factors polynomial in problem parameters β, G and D. By comparison, Garber et al. [8] obtain a regret bound of $O((dG^2/\lambda^3)\log T)$ w.h.p. for this setting. Both of our algorithms completely remove the dimension factor d, and reduce scaling w.r.t. the strong convexity parameter by a factor of $1/\lambda^2$ and $1/\lambda$ respectively. Moreover, their results require the losses to have a Lipschitz Hessian, an assumption we do not make.

In addition, we consider the case that F is convex (but not strongly convex), and apply our above results by means of regularization. As a corollary of the first theorem, we obtain for this setting regret that scales as $\widetilde{O}(\sqrt{T})$, matching up to logarithmic and additive factors the optimal rate for the *individually* convex setting. Similarly, the second theorem implies a $\widetilde{O}(T^{2/3})$ regret algorithm which is also memory efficient for the convex F case. Notably, a similar reduction applied to the results of Garber et al. [8] would yield a regret bound of $\widetilde{O}(dT^{3/4})$. This highlights the significance of our improvement to the dependence on λ.

1.2 Overview of techniques

Our approach builds on the observation that regret on a random order loss sequence may be expressed as the average generalization error w.r.t a without-replacement training sample. In light of this, we relate random order regret to a suitable notion of algorithmic stability, mimicking in a sense a well known argument previously employed in the context of i.i.d. sampled training sets [3, 21]. At a high level, stability measures the sensitivity of an algorithm to small changes in its training set, and is a classical approach to proving generalization bounds [3, 21]. More often than not, the particular notion used in practice is *uniform* stability, where sensitivity is measured w.r.t. the worst case small change in the training dataset; $\max_{S\subseteq\mathcal{Z}(m)} \|f(\mathcal{A}(S) - f(\mathcal{A}(S')))\|$.

Our key insight is that while we cannot hope for uniform stability as losses are not assumed to be convex individually, we may exploit strong convexity of the population loss to show SGD admits *average* stability; $\mathbb{E}_{S\sim\mathcal{Z}(m)} \|f(\mathcal{A}(S)) - f(\mathcal{A}(S'))\| \le \epsilon(m)$. In particular, we prove the gradient update is *contractive in expectation*; $\mathbb{E}\,\|x - \eta\nabla f(x) - (y - \eta\nabla f(y))\| \le \mathbb{E}\,\|x - y\|$ under the cumulative

strong convexity assumption. In turn, this yields a stability result which implies regret that scales with $1/t$ for the early rounds up to $t \approx T/\kappa$, where κ is the condition number of the problem. In short, as the game progresses the bias of the random order gradient estimates increases, and the gradient update becomes unstable.

To overcome the loss of stability in later rounds, we devise a simple online sampling mechanism that generates i.i.d. uniform samples from a random order distribution, effectively ensuring unbiased gradient estimates throughout all T rounds, and consequently optimal regret up to additive factors. Finally, our approach allows us to develop an analysis framework that naturally accommodates SGD based algorithms in the random order model, and in a broader sense establish stability of SGD in a new, relatively general setting.

1.3 Related work

Random-Order Online Optimization was proposed in the recent work of [8] (where it is referred to as ROOCO), who establish an $O((dG^2/\lambda^3)\log T)$ regret upper bound for G-Lipschitz λ-cumulative-strongly convex losses. In the classical OCO setup [24, 12], under the assumption the losses are λ-strongly convex individually, it is well known the minimax regret scales as $\Theta((G^2/\lambda)\log T)$, and that the lower bound also applies under the assumption the adversary is i.i.d. stochastic [14, 13]. In addition, it is well known (see e.g., [4]) that the upper bound for the i.i.d. adversary is obtained in expectation by SGD also for non-convex losses, as long as the expected loss is strongly convex. In this work, we show that the same regret upper bound also holds for the random order adversary.

Also relevant to our work is the study of stability and generalization [3, 21] in modern learning theory, and in particular stability properties of SGD. Proving generalization bounds with stability arguments is a well known approach dating back at least to [19, 5, 6]. The specific notion of average stability in the i.i.d. setting we draw upon was defined in [21], though many similar measures have appeared in the literature long before their work (see [3, 21] for an overview). The influential work of [11] gave the first generalization bounds for general forms of SGD with an analysis relying on the notion of uniform stability [3]. Since then, several works have used a similar approach to gain further insight into stability and generalization properties of SGD, e.g., [15, 7, 2].

Finally, a related line of work [9, 22, 10, 16, 20, 18] studies SGD without-replacement for solving finite-sum optimization problems, a setting commonly encountered in offline machine learning applications. Here, multiple epochs of SGD are executed over a given training set, with the objective to produce a single output (approximately) minimizing the expected loss. However, the majority of the results are obtained under the (vastly simplifying) assumption that the individual loss functions are convex, and therefore do not apply in our setting. In addition, the performance metric of interest is convergence rate, and not regret which is the focus of our paper. Recently, a number of papers have tried to relax the convexity assumption [10, 17, 1, 18], but the bounds they obtain are under conditions inapplicable for our setting. Specifically, they impose a requirement that the number of epochs passes a certain threshold strictly larger than one. Moreover, state-of-the-art bounds achieved by [1] are suboptimal w.r.t ours even had we ignored the epoch requirement.

2 Setup: Random-Order Online Optimization

In this section, we review notation and assumptions used throughout the paper, and give the formal definition of the model we consider. We let $Z = \{\zeta_1, \dots, \zeta_T\}$ denote an arbitrary set of T different datapoints, and denote by $W \subseteq \mathbb{R}^d$ a closed convex set with diameter bounded by $D := \max_{x,y \in W} \|x - y\|$. In addition, we let $\Pi(x) := \Pi_W(x) := \arg\min_{w \in W} \|x - w\|^2$ denote the orthogonal projection onto W.

We consider a loss function $f : W \times Z \to \mathbb{R}$, and denote the average (also expected / population) loss by $F(w) := \frac{1}{T}\sum_{z \in Z} f(w; z)$. We make the following assumptions;

Assumption 1 (Individual Lipschitz Continuity)**.** For all $z \in Z$, $f(\cdot\,; z)$ is G-Lipschitz; $\|\nabla f(w; z)\| \leq G \ \forall w \in W$.

Assumption 2 (Individual smoothness)**.** For all $z \in Z$, $f(\cdot\,; z)$ is β-smooth; $\|\nabla f(x; z) - \nabla f(y; z)\| \leq \beta\|x - y\| \ \forall x, y \in W$.

Assumption 3 (Cumulative strong convexity)**.** F is λ-strongly convex; $\nabla^2 F(w) \succeq \lambda I \ \forall w \in W$.

In addition, we define the condition number of F by $\kappa := \beta/\lambda$.

We will be primarily interested in the random sequence of losses $f(\cdot; z_t)$ obtained from uniformly random orderings of Z. Let $z_1, \ldots, z_m \sim \mathfrak{Z}(m)$ denote a random sequence of m datapoints, where $z_t = \zeta_{\sigma_t} \in Z$ and $\sigma : [T] \to [T]$ is a uniformly random permutation. Equivalently, $\mathfrak{Z}(m)$ may be also considered as the distribution of m datapoints sampled sequentially without-replacement from Z. Omitting the number of samples parameter and writing $z \sim \mathfrak{Z}$ denotes a uniformly random sample of a single datapoint from Z. In addition, if $S = (z_1, \ldots, z_m)$ is a sequence of datapoints, $z \sim \mathfrak{Z} \setminus S$ denotes a uniformly random sample of a single datapoint from $Z \setminus S$. In sake of conciseness, we write $S, \tilde{z} \sim \mathfrak{Z}(m, 1)$ to denote a sample of a sequence $S \sim \mathfrak{Z}(m)$, followed by a sample from the complement $\tilde{z} \sim \mathfrak{Z} \setminus S$. This, of course, is equivalent to sampling $z_1, \ldots z_{m+1} \sim \mathfrak{Z}(m+1)$, and then setting $S = (z_1, \ldots, z_m)$ and $\tilde{z} = z_{m+1}$.

Given a random order sequence $z_1, \ldots, z_T \sim \mathfrak{Z}(T)$, we consider the problem of minimizing the expected regret with an online algorithm. We denote the minimizer of the population loss F by $w^* := \arg\min_{w \in W} F(w)$, and let w_t denote the iterates produced by an online algorithm $\mathcal{A}$. The expected regret of $\mathcal{A}$ on $\mathfrak{Z}(T)$ is defined as;

$$\mathcal{R}_T := \mathbb{E}\Big[\sum_{t=1}^{T} f(w_t; z_t) - f(w^*; z_t)\Big],$$

where the expectation is over the random order sequence and any randomness potentially introduced by $\mathcal{A}$. Finally, when a sequence of realized datapoints $z_1, \ldots, z_m$ is clear from context, we let $F_m(w) := \frac{1}{m}\sum_{t=1}^{m} f(w; z_t)$ denote their empirical average loss.

3 Stability and Generalization

In this section, we discuss notions of stability and generalization when sampling without-replacement, and give basic results relating to stability of SGD in the setting under consideration. We work with ordered training sets, and write $S = (z_1, \ldots, z_m)$ to make the ordering explicit in our notation. When such a training set is in context along with another datapoint $\tilde{z}_i$, we define $S^{(i)} := (z_1, \ldots, z_{i-1}, \tilde{z}_i, z_{i+1}, \ldots, z_m)$ to be the new training set formed by taking S and swapping the i'th datapoint z_i with $\tilde{z}_i$. Finally, we say $\mathcal{A}$ is a learning algorithm if it maps training sets of any length to a decision; $\mathcal{A} : Z^* \to W$.

3.1 Recap: Stability and generalization in the i.i.d. setting

In this section we recall the relvant definitions previously studied in the i.i.d. setting. Here, we assume the training set $S = (z_1, \ldots, z_m) \sim \mathcal{D}^m$ is an i.i.d. sample of m datapoints from some predefined distribution $\mathcal{D}$ over elements of Z.

Definition (on-average generalization; [21, 3])**.** We say a learning algorithm $\mathcal{A}$ on-average-generalizes with rate $\epsilon_{\text{gen}}(m)$ if for all m;

$$|\,\mathbb{E}_{S\sim\mathcal{D}^m}[F_m(\mathcal{A}(S)) - F(\mathcal{A}(S))]| \le \epsilon_{\text{gen}}(m).$$

The definition of stability that follows relates a small change in the training set $S \to S^{(i)}$ to the change in the learning algorithm's performance. Here, as one would expect, the swapped datapoint $\tilde{z}_i \sim \mathcal{D}$ is sampled independently from the original training set sample S.

Definition (average-RO stability; [21, 3])**.** We say a learning algorithm $\mathcal{A}$ is average-replace-one stable with rate $\epsilon_{\text{stab}}(m)$ if for all m;

$$\left|\frac{1}{m}\sum_{i=1}^{m} \mathbb{E}_{S\sim\mathcal{D}^m, \tilde{z}_i\sim\mathcal{D}}[f(\mathcal{A}(S); \tilde{z}_i) - f(\mathcal{A}(S^{(i)}); \tilde{z}_i)]\right| \le \epsilon_{\text{stab}}(m).$$

With the above definitions, it is well known stability and generalization are in fact equivalent (e.g., [21]).

3.2 Stability and generalization without replacement

In this section we discuss the analogous notions suitable for sampling without-replacement. We adopt the term out-of-sample (oos) to distinguish the without-replacement setting, and say $\mathcal{A}$ on-average-generalize-oos with rate $\epsilon_{\text{gen}}(m)$ if

$$\left|\mathbb{E}_{S,\tilde{z}\sim\mathfrak{Z}(m,1)}\left[f(\mathcal{A}(S); \tilde{z}) - F_m(\mathcal{A}(S))\right]\right| \le \epsilon_{\text{gen}}(m). \tag{1}$$

Note that here, generalization is measured w.r.t. a datapoint drawn out-of-sample, and in particular *not independently* of S. The situation is similar for the notion of stability; while in the i.i.d. case the "non-coupled" index i in $S^{(i)}$ hosts a different datapoint sampled independently, here this datapoint is sampled from the complement $Z \setminus S$. The analogous definition for stability without-replacement says a learning algorithm $\mathcal{A}$ is average-replace-one-oos stable with rate ϵ_{stab} if

$$\left| \frac{1}{m} \sum_{i=1}^{m} \mathbb{E}_{S,\tilde{z}_i \sim \mathfrak{Z}(m,1)} \left[f(\mathcal{A}(S); \tilde{z}_i) - f(\mathcal{A}(S^{(i)}); \tilde{z}_i) \right] \right| \leq \epsilon_{\text{stab}}(m). \tag{2}$$

However, it will be more convenient in our case to work with a slightly different definition, which relates to the distance between *outputs* of the learning algorithm, rather than to the change in out-of-sample loss. We consider w.l.o.g. *randomized* learning algorithms $A : Z^* \times \mathcal{X} \to W$, where $\xi \in \mathcal{X}$ denotes the internal random seed used by $\mathcal{A}$. For convenience, we slightly overload notation and let $\xi \sim \mathcal{X}$ denote the distribution over $\mathcal{A}$'s random seeds.

Definition 1 (on-average-oos stability)**.** A learning algorithm $\mathcal{A}$ is on-average-oos stable with rate $\epsilon_{\text{stab}}(m)$ on random order distribution $\mathfrak{Z}$ if

$$\max_{i \leq m} \mathbb{E}_{S,\tilde{z}_i \sim \mathfrak{Z}(m,1), \xi \sim \mathcal{X}} \left[\|\mathcal{A}(S;\xi) - \mathcal{A}(S^{(i)};\xi)\| \right] \leq \epsilon_{\text{stab}}(m).$$

We wish to draw the reader's attention to two important aspects of the above definition. First, note that the measure is w.r.t. *random* training sets, which significantly differs from uniform stability where worst case training sets are considered. The maximum in the definition relates to the *index* of the swapped sample, and not to the training sets $S, S^{(i)}$. Second, the same random seed ξ is fed to $\mathcal{A}$ on both training sets, that is, we measure the expected distance between outputs subject to a maximal coupling of the algorithm's randomness.

When we discuss an *online* algorithm $\mathcal{A}$ and a random order sequence $z_1, \ldots, z_T \sim \mathfrak{Z}(T)$ is in context, we denote by $z_{1:m} = (z_1, \ldots, z_m)$ the prefix of length m. In addition, if $w_{m+1} = \mathcal{A}(z_{1:m}; \xi)$, we denote the coupled iterate by

$$w_{m+1}^{(i)} := \mathcal{A}(z_{1:m}^{(i)}; \xi) = \mathcal{A}(z_1, \ldots, z_{i-1}, \tilde{z}_i, z_{i+1}, \ldots, z_m; \xi), \quad \text{where } \tilde{z}_i = z_{m+1}. \tag{3}$$

With this notation, if $\mathcal{A}$ satisfies Definition 1 with rate $\epsilon(m)$, we have $\mathbb{E}\,\|w_{m+1} - w_{m+1}^{(i)}\| \leq \epsilon(m)$ for all $i \leq m$. Note that an online learning algorithm is nothing more than a learning algorithm that respects the order of the samples it is given as input.

We conclude this section with the lemma below, relating the population and out-of-sample performance gap to stability of the learning algorithm as defined above. The proof hinges on equivalence of stability and generalization in the without-replacement setting, and is provided in the supplementary.

Lemma 1. *Assume $\mathcal{A}$ is on-avg-oos stable with rate $\epsilon(m)$, and let $\ell : W \times Z \to \mathbb{R}$ be any L-Lipschitz loss function. Then;*

$$\mathbb{E}_{S,\tilde{z} \sim \mathfrak{Z}(m,1), z \sim \mathfrak{Z}} \left[\ell(\mathcal{A}(S); \tilde{z}) - \ell(\mathcal{A}(S); z) \right] \leq \frac{Lm}{T} \epsilon(m).$$

If $\mathcal{A}$ is an online algorithm producing iterates w_t and $z_1, \ldots, z_t \sim \mathfrak{Z}(t)$ a random order sample, this immediately implies

$$\mathbb{E}[f(w_t; z_t) - F(w_t)] \leq \frac{G(t-1)}{T} \epsilon(t-1).$$

We note that a similar implication appeared in [16], although the connection to algorithmic stability was not clearly identified.

3.3 Average stability of SGD

In this section, we develop the basic tools employed to establish that under appropriate conditions, SGD is algorithmically stable in the sense of Definition 1. Crucially, by considering *average* stability we are able to leverage strong convexity of the expected function F and prove the desired result. For $w \in W$ and $\psi : W \to \mathbb{R}$, we denote by $\mathcal{G}(w; \psi, \eta) = \Pi(w - \eta \nabla \psi(w))$ a projected gradient descent step from w. Our key lemma stated below, says a gradient step on a random function ψ is contractive in expectation when ψ is strongly convex in expectation, and the step-size is sufficiently small. The proof follows from standard arguments, and is deferred to the supplementary.

Lemma 2. *Consider an arbitrary distribution $\mathcal{P}$ of G-Lipschitz and β-smooth functions $\psi : W \to \mathbb{R}$ such that $\Psi(w) \coloneqq \mathbb{E}\,\psi(w)$ is μ-strongly-convex. Then for any $x, y \in W$, a gradient descent step with step-size $\eta \le \mu/\beta^2$ satisfies;*

$$\mathbb{E}_{\psi\sim\mathcal{P}}\,\|\mathcal{G}(x;\psi,\eta) - \mathcal{G}(y;\psi,\eta)\| \le \Big(1 - \frac{\eta\mu}{2}\Big)\|x - y\|.$$

Notice that the above result dictates for the step-size to be lesser than μ/β^2, which is roughly a factor of $1/\kappa$ smaller than needed to ensure (deterministic) contractivity under the assumption of individually strongly convex losses (see [11]). Lemma 2 serves as a building block to prove stability of SGD subject to relatively generic conditions, which we do next. Loosely speaking, if two training sequences do not differ too much, and the conditional expected loss is strongly convex, the expected distance between SGD iterates shrinks proportionally to the number of iterations executed.

We denote by $\mathrm{GD}(S; w_1, m, \{\eta_t\})$ the iterate $w_{m+1} \in W$ produced by executing m projected gradient descent steps on a sequence of datapoints $S = (z_1, \ldots, z_\tau)$, $\tau \ge m$, starting at the initial point $w_1 \in W$, with step-sizes $\{\eta_t\}$. When any one of w_1, m or $\{\eta_t\}$ are clear from context, they may be omitted in sake of conciseness. Our next lemma quantifies how small perturbations in random training sets translate to the expected change in outputs of SGD.

Lemma 3. *Let $i \le m \in \mathbb{N}$, and $S = (z_1, \ldots, z_m), S' = (z'_1, \ldots, z'_m)$ be two random datapoint sequences. Further, assume that for $0 < \mu$ and $0 \le \delta \le \mu/2\beta$, it holds that*

(i) $\Pr(z_t \ne z'_t \mid \mathcal{F}_{t-1}) \le \delta$ *for all* $t \ne i$*;*

(ii) $\mathbb{E}\left[f(w; z_t) \mid \mathcal{F}_{t-1}\right]$ *is μ-strongly convex as a function of $\mathcal{F}_{t-1}$-measurable $w \in W$,*

where $\mathcal{F}_{t-1} \coloneqq (z_1, z'_1, \ldots, z_{t-1}, z'_{t-1})$. Then, for step-size schedule $\eta_t = \min\left\{\tilde{\mu}/\beta^2, 2/\tilde{\mu}t\right\}$ with $\tilde{\mu} \coloneqq \mu - \delta\beta$, and any $w_1 \in W$, we have;

$$\mathbb{E}\,\|\mathrm{GD}(S; w_1, m) - \mathrm{GD}(S'; w_1, m)\| \le \frac{4G}{\tilde{\mu}m}(1 + 4\delta m).$$

The proof of the above lemma is given by following the recursive relation specified by Lemma 2 and the conditions imposed on the random sequences S, S'. The details are rather technical and are thus deferred to the supplementary. To conclude this section, we state and prove a simple corollary of Lemma 3, establishing stability of SGD when the number of steps taken is sufficiently small.

Corollary 1. *Let $w_1 \in W$, and set $\eta_t = \min\left\{\lambda/2\beta^2, 4/\lambda t\right\}$. Then the SGD update defined by $w_{t+1} = \Pi(w_t - \eta_t \nabla f(w_t; z_t))$ is on-avg-oos stable with rate*

$$\epsilon_{stab}(m) \le \frac{8G}{\lambda m},$$

for all $m \le T/2\kappa$.

Proof. Fix $i \le m$, let $\mathcal{A}$ denote the SGD algorithm, and consider a random order sequence $z_1, \ldots, z_T \sim \mathfrak{Z}(T)$. We have $w_{m+1} = \mathcal{A}(z_{1:m})$, and $w^{(i)}_{m+1} = \mathcal{A}(z^{(i)}_{1:m})$ as defined in Eq. (3) (note that here though, $\mathcal{A}$ has no internal randomness). Next, we verify conditions for Lemma 3 are satisfied with the two sequences $S \coloneqq z_{1:m}$ and $S' \coloneqq (z'_1 \ldots, z'_m) \coloneqq z^{(i)}_{1:m}$.

Let $\mathcal{F}_{t-1} \coloneqq (z_1, z'_1, \ldots, z_{t-1}, z'_{t-1})$, and by definition of S and S', we have that $z_t = z'_t$ for all $t \ne i$, hence clearly $\Pr(z_t \ne z'_t \mid \mathcal{F}_{t-1}) = 0$ for all $t \ne i$. For the second condition, let F_{t-1} denote the set content of $\mathcal{F}_{t-1}$ hosting all datapoints observed prior to round t; $\mathsf{F}_{t-1} \coloneqq \{z \in Z \mid z \in \mathcal{F}_{t-1}\}$. This means F_{t-1} contains exactly $\{z_1, \ldots, z_{t-1}\}$, and perhaps $\tilde{z}_i$ depending on whether $t > i$, hence $k \coloneqq |\mathsf{F}_{t-1}| \in \{t, t-1\}$. Now, given $\mathcal{F}_{t-1}$, we have that z_t is uniform over $\mathfrak{Z} \setminus \mathsf{F}_{t-1}$, therefore

$$\mathbb{E}[f(w; z_t) \mid \mathcal{F}_{t-1}] = \frac{1}{T-k}\sum_{z\in Z\setminus\mathsf{F}_{t-1}} f(w; z) = \frac{T}{T-k}\Big(F(w) - \frac{1}{T}\sum_{z\in\mathsf{F}_{t-1}} f(w; z)\Big).$$

By our smoothness assumption, $\frac{1}{T}\sum_{z\in\mathsf{F}_{t-1}} f(w; z)$ is $(k\beta/T)$-smooth, and in addition $k\beta/T \le m\beta/T \le (\lambda/2\beta)\beta = \lambda/2$. Therefore, by λ-strong convexity of F we get that the last term in the above derivation is at least $\lambda/2$-strongly-convex. Therefore, by Lemma 3 with $\mu \coloneqq \lambda/2, \delta \coloneqq 0$ it now follows that

$$\mathbb{E}\,\|w_{m+1} - w^{(i)}_{m+1}\| = \mathbb{E}\,\|\mathrm{GD}(S) - \mathrm{GD}(S')\| \le \frac{8G}{\lambda m},$$

which completes the proof. □

4 SGD for Random Order Online Optimization

In this section, we present two algorithms for random order online optimization. Lemma 1 motivates us to derive SGD based algorithms that obtain low regret w.r.t. the population loss F, *and* are algorithmically stable in the sense of Definition 1. Given a random order sequence $z_1, \ldots, z_T \sim \mathfrak{Z}(T)$, the SGD update with gradient estimates $\{\hat{g}_t\}$ and step sizes $\{\eta_t\}$ is given by

$$w_{t+1} \leftarrow \Pi(w_t - \eta_t \hat{g}_t).$$

It is not hard to show that the regret w.r.t. F of SGD is directly related to the error terms introduced by using $\hat{g}_t$ in place of the true population loss gradients $\nabla F(w_t)$. This fact is made formal in the lemma below, which serves as a starting point for the analysis of both algorithms we present. The proof follows from standard arguments, and is deferred to the supplementary.

Lemma 4. *Consider τ iterations of SGD with gradient estimates $\{\hat{g}_t\}$ and step-size schedule $\eta_t = \min\left\{\tilde{\mu}/\beta^2, 2/(\tilde{\mu}t)\right\}$. We have that the following bound holds with probability one;*

$$\sum_{t=1}^{\tau} F(w_t) - F(w^*) \leq \frac{\beta^2 D^2}{2\tilde{\mu}} + \frac{G^2}{\tilde{\mu}}(1 + \log \tau) + \sum_{t=1}^{\tau} (\nabla F(w_t) - \hat{g}_t)^{\mathsf{T}}(w_t - w^*). \tag{4}$$

4.1 Reservoir SGD

In light of Corollary 1, it is evident that a different strategy is necessary to achieve stability in late rounds of the online game. As a solution, Algorithm 1 presented here employs a sampling procedure reminiscent of reservoir sampling [23], which results in gradient estimates $\hat{g}_t$ that are *conditionally unbiased* estimates of $\nabla F(w_t)$ throughout all T rounds. This ensures the conditionally expected function on every round is strongly convex, thereby implying stability is maintained for the duration of the game. As another implication, the error terms on the RHS of Eq. (4) vanish, which essentially reduces the optimization problem (i.e., regret w.r.t. the population loss F) to the i.i.d. setting. Consequently, a regret bound will follow from a batch-to-online conversion supported by Lemma 1 and Lemma 3.

Algorithm 1 ReservoirSGD

1: **input:** step-sizes $\eta_1, \ldots, \eta_T \in \mathbb{R}_+$, $w_1 \in W$
2: **for** $t = 1$ to T **do**
3: Play w_t, Observe z_t
4: Set $\hat{g}_t := \nabla f(w_t; z'_t)$, where $z'_t = \begin{cases} z_t & \text{w.p. } 1 - \frac{t-1}{T} \\ \mathrm{Unif}(z_1, \ldots, z_{t-1}) & \text{w.p. } \frac{t-1}{T} \end{cases}$
5: $w_{t+1} \leftarrow \Pi(w_t - \eta_t \hat{g}_t)$
6: **end for**

Our first lemma given below, shows that the intermediate sequence of datapoints z'_t generated in line 4 are i.i.d. uniformly distributed, which immediately implies that $\mathbb{E}\,\hat{g}_t = \nabla F(w_t)$. The proof is standard and deferred to the supplementary.

Lemma 5. *The $\{z'_t\}$ intermediate sequence produced by Algorithm 1 in line 4 is uniform over Z and i.i.d.;*

$$\forall z \in Z; \quad \Pr(z'_t = z) = \Pr(z'_t = z \mid z_{<t}) = \tfrac{1}{T}.$$

Next, we argue Algorithm 1 maintains average stability with the desired rate throughout all T rounds.

Lemma 6. *Assume $T \geq 2\beta/\lambda$, and set $\tilde{\mu} := \lambda - \beta/T$. Then Algorithm 1 with step-size schedule $\eta_t = \min\left\{\tilde{\mu}/\beta^2, 2/(\tilde{\mu}t)\right\}$ is on-avg-oos stable (Definition 1) with rate*

$$\epsilon_{stab}(m) \leq \frac{40G}{\lambda m}.$$

Proof. Let $i \leq m \leq T$, and recall the definition of the coupled iterate in Eq. (3). We have that

$$\mathbb{E}\,\|w_{m+1} - w^{(i)}_{m+1}\| = \mathbb{E}\,\|\mathrm{GD}(S') - \mathrm{GD}(S'')\|,$$

where $S' = (z'_1, \ldots, z'_m)$ and $S'' = (z''_1, \ldots, z''_m)$ denote the intermediate sequences (line 4) produced when running Algorithm 1 on $z_{1:m}$ and $z^{(i)}_{1:m}$ respectively. Now, consider the indexes of datapoints

selected by the sampling mechanism on line 4 which we denote by j_t, meaning $j_t = l$ when $z'_t = z_l$. Since the same random seed is used for each coupled iterate, we have that the indexes j_t from both execution paths are the same. Next, we will show S', S'' satisfy the conditions of Lemma 3.

Indeed, denote by $\mathcal{F}_t$ the filtration encapsulating all randomness (random order and algorithm) up to and including round t. Then by Lemma 5 and since $z_{1:m}$ and $z^{(i)}_{1:m}$ differ only at index i, we have for any $t > i$;

$$\Pr(z'_t \neq z''_t \mid \mathcal{F}_{t-1}) = \Pr(j_t = i \mid \mathcal{F}_{t-1}) = \tfrac{1}{T}.$$

In addition, it trivially follows that $\Pr(z'_t \neq z''_t \mid \mathcal{F}_{t-1}) = 0$ for all $t < i$. Owed to our assumption on T, we have $1/T \leq \lambda/2\beta$, and so the first condition is satisfied. For the second condition, note that again by Lemma 5, $\mathbb{E}[f(w; z'_t) \mid \mathcal{F}_{t-1}] = F(w)$ for any $\mathcal{F}_{t-1}$-measurable $w \in W$, which immediately implies λ-strong convexity of the conditionally expected function.

By the above, we obtain that S', S'' follow a distribution satisfying conditions required by Lemma 3 with $\mu := \lambda, \delta := 1/T$, therefore,

$$\mathbb{E}\,\|\mathrm{GD}(S'; m) - \mathrm{GD}(S''; m)\| \leq \frac{4G}{(\lambda - (1/T)\beta)m}\left(1 + 4\frac{m}{T}\right) \leq \frac{40G}{\lambda m},$$

and we are done. □

A regret bound for Algorithm 1 readily follows, as we have essentially established both convergence rate and stability of the algorithm. Below, we state and prove our main result concluding this section.

Theorem 1. *Running Algorithm 1 with step-size schedule* $\eta_t = \min\left\{\tilde{\mu}/\beta^2, 2/(\tilde{\mu}t)\right\}$ *where* $\tilde{\mu} := \lambda - \beta/T$*, it is guaranteed that;*

$$\mathbb{E}\left[\sum_{t=1}^{T} f(w_t; z_t) - f(w^*; z_t)\right] \leq \frac{2G^2}{\lambda}(1 + \log T) + \frac{40G^2 + \beta^2 D^2 + 2\beta G D}{\lambda}$$

Proof. By Lemma 5, we have that for all $t \leq T$,

$$\mathbb{E}[\nabla f(w_t; z'_t))^{\mathsf{T}}(w_t - w^*)] = \mathbb{E}[\mathbb{E}_t[\nabla f(w_t; z'_t)])^{\mathsf{T}}(w_t - w^*)] = \mathbb{E}[\nabla F(w_t)^{\mathsf{T}}(w_t - w^*)],$$

where $\mathbb{E}_t[\cdot] = \mathbb{E}[\,\cdot \mid z_1, z'_1, \ldots, z_{t-1}, z'_{t-1}]$ denotes the conditional expectation w.r.t. all rounds up to and not including t. This implies the gradient error terms on the RHS of Eq. (4) vanish. In addition, note we may assume $T \geq 2\beta/\lambda$, for otherwise it trivially follows that $\sum_{t=1}^{T} f(w_t; z_t) - f(w^*; z_t) \leq GDT \leq \frac{2\beta G D}{\lambda}$. Hence $\delta := 1/T \leq \lambda/2\beta$, and by Lemma 4 we obtain the following bound on the regret w.r.t. the population loss F;

$$\mathbb{E}\left[\sum_{t=1}^{T} F(w_t) - F(w^*)\right] \leq \frac{\beta^2 D^2}{\lambda} + \frac{2G^2}{\lambda}(1 + \log T).$$

To establish online performance, observe that by Lemma 1 and Lemma 6 we have

$$\mathbb{E}[f(w_t; z_t) - F(w_t)] \leq \frac{G(t-1)}{T}\epsilon_{\mathrm{stab}}(t-1) = \frac{G(t-1)}{T}\frac{40G}{\lambda(t-1)} = \frac{40G^2}{\lambda T}.$$

Therefore,

$$\begin{aligned}\mathbb{E}\left[\sum_{t=1}^{T} f(w_t; z_t) - f(w^*; z_t)\right] &= \sum_{t=1}^{T}\mathbb{E}[f(w_t; z_t) - F(w_t)] + \mathbb{E}\left[\sum_{t=1}^{T} F(w_t) - F(w^*)\right] \\ &\leq \frac{40G^2}{\lambda} + \frac{\beta^2 D^2}{\lambda} + \frac{2G^2}{\lambda}(1 + \log T),\end{aligned}$$

and the result follows. □

4.2 SGD without replacement

While Algorithm 1 obtains regret which is optimal up to additive factors, it is memory intensive due to the sampling procedure requiring the history of the entire loss sequence. This motivates Algorithm 2 which uses the random order gradients as they arrive, and sacrifices a factor of κ in the regret bound for lower memory requirements.

Algorithm 2 SGD-without-replacement

input: $\tau, T \in \mathbb{N}$, step-sizes $\eta_1, \ldots, \eta_\tau \in \mathbb{R}_+$, $w_1 \in W$
for $t = 1$ to τ **do**
  Play w_t, Observe z_t
  $w_{t+1} \leftarrow \Pi(w_t - \eta_t \hat{g}_t)$, where $\hat{g}_t := \nabla f(w_t; z_t)$
end for
for $t = \tau + 1, \ldots, T$: play $w_t \equiv \bar{w} := \frac{1}{\tau}\sum_{i=1}^{\tau} w_i$.

With Corollary 1, we already know Algorithm 2 is stable with rate $O(1/m)$ for $\tau \leq T/2\kappa$, at least for all iterates excluding $\bar{w}$. However, it is not clear a priori whether these iterates also obtain good convergence rate—to prove this we must control the gradient error terms on the RHS of Eq. (4). Evidently, with random order gradient estimates the behavior of these error terms is also related to stability in the same sense that the actual losses are. By Lemma 1 with $\ell(w; z) := \nabla f(w; z_t)^\top (w - w^*)$, it only remains to derive the appropriate Lipschitz constant, which is done in our next lemma. Notably, this means that the two sources of error, gradient estimates and the batch-to-online gap, both hinge on the very same stability property of the algorithm.

Lemma 7. *Running Algorithm 2 with a step-size schedule $\eta_t = \min\left\{\lambda/2\beta^2, 4/\lambda t\right\}$ and $\tau = T/2\kappa$, we have that the gradient error terms on the RHS of Eq. (4) are bounded, for all $t \leq \tau$, as*

$$\mathbb{E}[(\nabla F(w_t) - \hat{g}_t)^\top (w_t - w^*)] \leq \frac{8G(G + \beta D)}{\lambda T}.$$

The proof of Lemma 7 is given by arguments following the above discussion, and is deferred to the supplementary. Next, we state our main theorem for this section providing the regret guarantees for Algorithm 2.

Theorem 2. *Running Algorithm 2 with a step-size schedule $\eta_t = \min\left\{\lambda/2\beta^2, 4/\lambda t\right\}$ and $\tau = T/2\kappa$, it holds that*

$$\mathbb{E}\left[\sum_{t=1}^{T} f(w_t; z_t) - f(w^*; z_t)\right] = O\left(\frac{\beta D G}{\lambda} + \frac{\beta^3 D^2}{\lambda^2} + \frac{\beta G^2}{\lambda^2}\log T\right).$$

The proof of Theorem 2, provided in the supplementary, hinges on the ingredients discussed thus far, along with the fact that averaging preserves stability.

4.3 The convex case

Given Algorithms 1 and 2, we can derive as an immediate corollary regret bounds for the case where the loss function f is only assumed to be convex in expectation, but not strongly convex. This is achieved by adding L2 regularization; pick $w_0 \in W$ arbitrarily, and consider $f^\alpha(w; z) := f(w; z) + \frac{\alpha}{2}\|w - w_0\|^2$. By transforming the gradient estimators $\hat{g}_t \leftarrow \hat{g}_t + \alpha(w_t - w_0)$ in both algorithms, we optimize for the regularized random order loss sequence $f^\alpha(\cdot; z_1), \ldots f^\alpha(\cdot; z_T)$, and obtain regret bounds for suitable choices of α. We state the result below, and provide its proof in the supplementary.

Corollary 2. *Assume the loss function $f : W \times Z \to \mathbb{R}$ is G-Lipschitz and β-smooth for all $z \in Z$ individually (Assumption 1 and Assumption 2), and that $\frac{1}{T}\sum_{z \in Z} f(\cdot; z)$ is convex over W. Then, we have the following guarantees for running algorithms 1 and 2 on the regularized loss sequence;*

(i) *For Algorithm 1 and $\alpha = 1/\sqrt{T}$;* $\mathbb{E}\left[\sum_{t=1}^{T} f(w_t; z_t) - f(w^*; z_t)\right] = \widetilde{O}(\sqrt{T})$,

(ii) *for Algorithm 2 and $\alpha = 1/T^{1/3}$;* $\mathbb{E}\left[\sum_{t=1}^{T} f(w_t; z_t) - f(w^*; z_t)\right] = \widetilde{O}(T^{2/3})$,

where big-$\widetilde{O}$ hides polynomial dependence on the problem parameters β, G, D, and logarithmic dependence on T.

We conclude by noting the above result underscores the importance of obtaining bounds with good dependence on the strong convexity parameter. In particular, only the optimal $1/\lambda$ dependence allows for regularization that guarantees optimal (up to logarithmic factors) performance w.r.t. in the non-strongly convex setting.

References

[1] K. Ahn, C. Yun, and S. Sra. Sgd with shuffling: optimal rates without component convexity and large epoch requirements. *arXiv preprint arXiv:2006.06946*, 2020.

[2] R. Bassily, V. Feldman, C. Guzmán, and K. Talwar. Stability of stochastic gradient descent on nonsmooth convex losses. *Advances in Neural Information Processing Systems*, 33, 2020.

[3] O. Bousquet and A. Elisseeff. Stability and generalization. *The Journal of Machine Learning Research*, 2:499–526, 2002.

[4] S. Bubeck et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

[5] L. Devroye and T. Wagner. Distribution-free inequalities for the deleted and holdout error estimates. *IEEE Transactions on Information Theory*, 25(2):202–207, 1979.

[6] L. Devroye and T. Wagner. Distribution-free performance bounds with the resubstitution error estimate (corresp.). *IEEE Transactions on Information Theory*, 25(2):208–210, 1979.

[7] V. Feldman and J. Vondrak. High probability generalization bounds for uniformly stable algorithms with nearly optimal rate. In *Conference on Learning Theory*, pages 1270–1279. PMLR, 2019.

[8] D. Garber, G. Korcia, and K. Levy. Online convex optimization in the random order model. In *International Conference on Machine Learning*, pages 3387–3396. PMLR, 2020.

[9] M. Gürbüzbalaban, A. Ozdaglar, and P. A. Parrilo. Why random reshuffling beats stochastic gradient descent. *Mathematical Programming*, pages 1–36, 2019.

[10] J. Haochen and S. Sra. Random shuffling beats sgd after finite epochs. In *International Conference on Machine Learning*, pages 2624–2633. PMLR, 2019.

[11] M. Hardt, B. Recht, and Y. Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International Conference on Machine Learning*, pages 1225–1234. PMLR, 2016.

[12] E. Hazan. Introduction to online convex optimization. *arXiv preprint arXiv:1909.05207*, 2019.

[13] E. Hazan and S. Kale. Beyond the regret minimization barrier: an optimal algorithm for stochastic strongly-convex optimization. In *Proceedings of the 24th Annual Conference on Learning Theory*, pages 421–436. JMLR Workshop and Conference Proceedings, 2011.

[14] E. Hazan, A. Agarwal, and S. Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

[15] B. London. A pac-bayesian analysis of randomized learning with application to stochastic gradient descent. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 2935–2944, 2017.

[16] D. Nagaraj, P. Jain, and P. Netrapalli. Sgd without replacement: Sharper rates for general smooth convex functions. In *International Conference on Machine Learning*, pages 4703–4711. PMLR, 2019.

[17] L. M. Nguyen, Q. Tran-Dinh, D. T. Phan, P. H. Nguyen, and M. van Dijk. A unified convergence analysis for shuffling-type gradient methods. *arXiv preprint arXiv:2002.08246*, 2020.

[18] S. Rajput, A. Gupta, and D. Papailiopoulos. Closing the convergence gap of sgd without replacement. In *International Conference on Machine Learning*, pages 7964–7973. PMLR, 2020.

[19] W. H. Rogers and T. J. Wagner. A finite sample distribution-free performance bound for local discrimination rules. *The Annals of Statistics*, pages 506–514, 1978.

[20] I. Safran and O. Shamir. How good is sgd with random shuffling? In *Conference on Learning Theory*, pages 3250–3284. PMLR, 2020.

[21] S. Shalev-Shwartz, O. Shamir, N. Srebro, and K. Sridharan. Learnability, stability and uniform convergence. *The Journal of Machine Learning Research*, 11:2635–2670, 2010.

[22] O. Shamir. Without-replacement sampling for stochastic gradient methods. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 46–54, 2016.

[23] J. S. Vitter. Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)*, 11(1):37–57, 1985.

[24] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pages 928–936, 2003.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] Our work is purely theoretical and has no negative societal impact.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]